



# Table of contents

---

## Overview

---

● Description	4
● Confidence	4
● Content	5

---

## Entities

---

● Malware	6
● Intrusion-Set	8
● Vulnerability	9
● Attack-Pattern	10
● Country	15
● Region	16
● Sector	17



## External References

- 
- External References

18

# Overview

## Description

Pikabot is a malware loader that was first observed in early 2023 and became very active following the takedown of Qakbot in August 2023. In December 2023, Pikabot activity ceased, possibly as a result of a new version of Qakbot that emerged. In February 2024, a new version of Pikabot was released with significant changes. The malware continues to pose a significant cyber threat and is in constant development, although the developers have decreased the complexity level of Pikabot's code by removing advanced obfuscation features.

## Confidence

*This value represents the confidence in the correctness of the data contained within this report.*

15 / 100

# Content

N/A

# Malware

## Name

Cobalt Strike - S0154

## Name

QakBot - S0650

## Name

Pikabot

## Name

QakBot

## Description

[QakBot](<https://attack.mitre.org/software/S0650>) is a modular banking trojan that has been used primarily by financially-motivated actors since at least 2007. [QakBot](<https://attack.mitre.org/software/S0650>) is continuously maintained and developed and has evolved from an information stealer into a delivery agent for ransomware, most notably [ProLock](<https://attack.mitre.org/software/S0654>) and [Egregor](<https://attack.mitre.org/software/S0554>). (Citation: Trend Micro Qakbot December 2020)(Citation: Red Canary Qbot) (Citation: Kaspersky QakBot September 2021)(Citation: ATT QakBot April 2021)

## Name

## Cobalt Strike

**Description**

[Cobalt Strike](<https://attack.mitre.org/software/S0154>) is a commercial, full-featured, remote access tool that bills itself as “adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors”. Cobalt Strike’s interactive post-exploit capabilities cover the full range of ATT&CK tactics, all executed within a single, integrated system.(Citation: cobaltstrike manual) In addition to its own capabilities, [Cobalt Strike](<https://attack.mitre.org/software/S0154>) leverages the capabilities of other well-known tools such as Metasploit and [Mimikatz](<https://attack.mitre.org/software/S0002>).(Citation: cobaltstrike manual)

# Intrusion-Set

## Name

BlackBasta



# Vulnerability

**Name**

CVE-2021-44228

**Description**

Apache Log4j2 contains a vulnerability where JNDI features do not protect against attacker-controlled JNDI-related endpoints, allowing for remote code execution.

# Attack-Pattern

Name
Obfuscated Files or Information
ID
T1027
Description

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](<https://attack.mitre.org/techniques/T1140>) for [User Execution](<https://attack.mitre.org/techniques/T1204>). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](<https://attack.mitre.org/techniques/T1027/010>) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](<https://attack.mitre.org/techniques/T1059>). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

**Name**

Phishing

**ID**

T1566

**Description**

Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a

trusted source, as well as evasive techniques such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages (e.g., [Email Hiding Rules](https://attack.mitre.org/techniques/T1564/008)).(Citation: Microsoft OAuth Spam 2022)(Citation: Palo Alto Unit 42 VBA Infostealer 2014) Another way to accomplish this is by forging or spoofing(Citation: Proofpoint-spoof) the identity of the sender which can be used to fool both the human recipient as well as automated security tools.(Citation: cyberproof-double-bounce) Victims may also receive phishing messages that instruct them to call a phone number where they are directed to visit a malicious URL, download malware,(Citation: sygnia Luna Month)(Citation: CISA Remote Monitoring and Management Software) or install adversary-accessible remote management tools onto their computer (i.e., [User Execution](https://attack.mitre.org/techniques/T1204)).(Citation: Unit42 Luna Moth)

**Name**

Process Injection

**ID**

T1055

**Description**

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

**Name**

Masquerading

**ID**

T1036

**Description**

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusible system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036). (Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](https://attack.mitre.org/techniques/T1090) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.

**Name**

Deobfuscate/Decode Files or Information

**ID**

T1140

**Description**

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file. (Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b`` command to reassemble binary fragments into a malicious payload. (Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/

encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016)

# Country

**Name**

Ukraine

**Name**

Russian Federation

# Region

**Name**

Eastern Europe

**Name**

Europe



# Sector

**Name**

Manufacturing

**Description**

Private entities transforming and selling goods, products and equipment which are not included in other activity sectors.

**Name**

Technologies

**Description**

Private entities related to the research, development, manufacturing and distribution of electronics, softwares, computers and products related to information technologies.

# External References

- 
- <https://www.zscaler.com/blogs/security-research/d-evolution-pikabot>
- 
- <https://otx.alienvault.com/pulse/65cc85c228ee9d9184a9f0c2>